7

A Note on Countercyclicity and Minimalist Grammars

Hans-Martin Gärtner and Jens Michaelis

7.1 Introduction

Minimalist grammars (MGs), as introduced in Stabler (1997), have proven a useful instrument in the formal analysis of syntactic theories developed within the minimalist branch of the principles–and–parameters framework (cf. Chomsky 1995, 2000). In fact, as shown in Michaelis (2001), MGs belong to the class of mildly context–sensitive grammars. Interestingly, without there being a rise in (at least weak) generative power, (extensions and variants of) MGs accommodate a wide variety of (arguably) "odd" items from the syntactician's toolbox, such as *head movement* (Stabler 1997, 2001), *affix hopping* (Stabler 2001), *(strict) remnant movement* (Stabler 1997, 1999), *adjunction* (Frey and Gärtner 2002), and (to some extent) *scrambling* (Frey and Gärtner 2002).[1]

Here, we would like to explore the possibility of enriching MGs with another controversial mechanism, namely, *countercyclic operations*. These operations allow structure building at any node in the tree instead of just at the root.[2] We will first discuss countercyclic ad-

[1] A strictly formal proof showing that at least the weak generative capacity is unaffected seems to be straightforward but is still outstanding for the corresponding formalizations of adjunction and scrambling. An empirically fully satisfactory MG–treatment of scrambling, however, requires further research.

[2] Note that affix hopping and head movement as formalized in the mentioned works can be considered to be countercyclic in the weaker sense that these operations

Proceedings of FGVienna: The 8th Conference on Formal Grammar.
Gerald Penn (ed.).

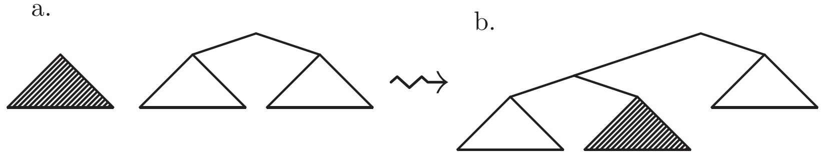
junction, which has repeatedly been postulated in the syntactic literature, especially in analyses of binding phenomena (Section 7.2.1). Then we sketch an extension of MGs that captures countercyclic adjunction (Section 7.2.2). As further discussed in Section 7.2.3, it turns out that, while weak and (even) strong generative capacity seem to remain essentially unaffected by this modification, there is an effect on what can be called *derivational generative power*, a category earlier introduced by Becker et al. (1992), which is considered to be “orthogonal” to the dimension of strong generative power. This is due to the fact that the latter is about derived structures while the former concerns derivation structures. In Section 7.3 we give an outlook on further variants of countercyclicity.

7.2 Countercyclic Adjunction

Let us briefly illustrate the crucial property of countercyclic operations, i.e. the capability of expanding the tree at a non–root position. Thus, a transition from (1a) to (1b) is countercyclic.

(1)

a. b.



7.2.1 Adjuncts and Binding

Countercyclic adjunction has been argued for among others by Lebeaux (1991) on the basis of contrasts like the following.

(2) a. *She$_i$ denied the claim that Mary$_i$ fell asleep
b. *She$_i$ liked the book that Mary$_i$ read
c. *Which claim that Mary$_i$ fell asleep did she$_i$ deny
d. Which book that Mary$_i$ read did she$_i$ like

Lebeaux’ account rests on the assumption that (2a) and (2b) are ruled out by Principle C of the Binding Theory (Chomsky 1981), according to which an R–expression like *Mary* must not be c–commanded by any coindexed constituent, such as *she* in our examples. The contrast in (2c)/(2d) would then follow, if there is a stage in the derivation of (2c) where such an illicit c–command relation holds, while there is no such stage in the derivation of (2d). Concretely put, (3a), i.e. the

simply do not lead to any “proper” tree expansion at any node.

stage before wh–movement applies to yield (3b), displays a Principle C violation.

(3) a. $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ deny $[_{DP}$ which claim $[_{CP}$ that Mary$_i$ fell asleep]]]]]
b. $[_{CP}[_{DP}$ which claim $[_{CP}$ that Mary$_i$ fell asleep]] $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ deny t]]]]

Crucially, things would be different for (2d), if instead of cyclic adjunction leading to stage (4a) and ultimately (4c) via the illicit stage (4b), adjuncts like relative clauses were allowed to be introduced "late," i.e. countercyclically. This alternative is illustrated in (5a)–(5c).

(4) a. $[_{DP}[_{DP}$ which book] $[_{CP}$ that Mary$_i$ read]]
⋮
b. $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ like $[_{DP}$ $[_{DP}$ which book] $[_{CP}$ that Mary$_i$ read]]]]]
c. $[_{CP}$ $[_{DP}$ $[_{DP}$ which book] $[_{CP}$ that Mary$_i$ read]] $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ like t]]]]

(5) a. $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ like $[_{DP}$ which book]]]]
b. $[_{CP}$ $[_{DP}$ which book] $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ like t]]]]
c. $[_{CP}$ $[_{DP}$ $[_{DP}$ which book] $[_{CP}$ that Mary$_i$ read]] $[_{C'}$ did $[_{IP}$ she$_i$ $[_{VP}$ like t]]]]

In derivation (5) there is no stage at which *Mary* is c–commanded by *she* and thus Principle C is complied with as desired.[3]

7.2.2 An MG–Treatment of Countercyclic Adjunction

An MG–account of countercyclic adjunction, subsuming the transition from (5b) to (5c) above, is straightforward. We have to adopt a variant of the *adjoin*–operation introduced into the MG–formalism by Frey and Gärtner (2002). This requires that in addition to *(basic) categorial features*, *m(erge)–selectors*, *(move–)licensees* and *(move–)licensors*, denoted in the form `x`, `=x`, `-x` and `+x`, respectively, an MG is equipped with *a(djoin)-selectors*, denoted in the form ≈`x`.

As for further notation and use of symbols: appearing within a given string of features, `#` serves to mark the substring of features to its right as "unchecked." `>` denotes "right constituent projects over left one," and `<` denotes "left constituent projects over right one."[4] Furthermore,

[3]We have to sidestep many ramifications of this account such as criticisms and alternatives in terms of reconstruction and copies, not to speak of alternative approaches to binding.

[4]Recall that a minimalist tree is always binary (branching), cf. Definition 15 from the appendix.

for minimalist trees τ, τ_1 and τ_2 such that τ_1 is a subtree of τ, $\tau\{\tau_1/\tau_2\}$ represents the result of replacing τ_1 by τ_2 in τ, and

(6) τ_1 is a *maximal projection* in τ in case it is identical to τ, or it is projected over by its sister constituent (i.e. in case each subtree of τ which is a proper supertree of τ_1 has a head other than the head of τ_1).

Assuming that υ and τ are minimalist trees such that υ displays $\approx\mathtt{x}$, and there is at least one maximal projection χ in τ fulfilling condition (7), *(right) adjunction* for the pair $\langle\upsilon,\tau\rangle$, i.e. (right) adjunction of υ to τ, can be defined as in (8).[5,6]

(7) The head–label of χ is of the form $\beta\mathtt{\#x}\beta'$ or $\beta\mathtt{x}\beta'\mathtt{\#}\beta''$ for some $\beta,\beta',\beta''\in \mathit{Feat}^*$.

(8) $\mathit{adjoin}(\upsilon,\tau)$

$$= \{\tau\{\chi/[_<\chi,\upsilon'\,]\}\,|\,\chi \text{ maximal projection in } \tau \text{ obeying (7)}\,\},$$

where υ' results from υ by interchanging the instances of `#` and $\approx\mathtt{x}$, the latter immediately following the former within the head–label of υ.[7]

In order to sketch our treatment of countercyclic adjunction in (5) we choose the following small MG–lexicon.[8]

(9)	a. `#.n.`*book*	b. `#.d.`*she*	c. `#.=d.v.`*like*
	d. `#.=n.d.-wh.`*which*	e. `#.=v.=d.i.`$\emptyset$	f. `#.=i.`$\approx$`d.`*that*
	g. `#.=i.+wh.c.`*did*	h. `#.i.`*Mary_read*	

(10a)–(10c) below correspond to (5a)–(5c), respectively.

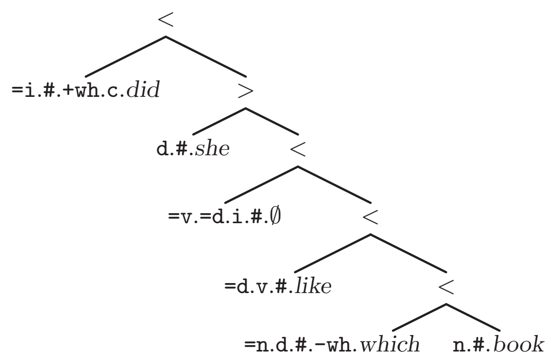
[5] Left adjunction can be defined analogously. Note that, in contrast with its counterpart in Frey and Gärtner (2002), *adjoin* as defined here does not necessarily map a pair of minimalist trees from its domain to a unique minimalist tree, there being potentially multiple "adjunction sites." The operation adopts the attractive type–preserving "x/x–approach" from *categorial grammar*. It should not be confused with the more general *(tree) adjoining* operation familiar from *tree adjoining grammar (TAG)* (cf. e.g. Joshi and Schabes 1997). The latter operation allows countercyclicity quite generally.

[6] A minimalist tree τ *displays* feature f if an instance of f starts the substring of unchecked features within τ's head–label. For any two minimalist trees υ and χ, $[_<\chi,\upsilon\,]$ (respectively, $[_>\chi,\upsilon\,]$) denotes the minimalist tree whose root immediately dominates subtrees χ and υ such that χ's root precedes υ's root, and such that χ's root projects over (respectively, is projected over by) υ's root.

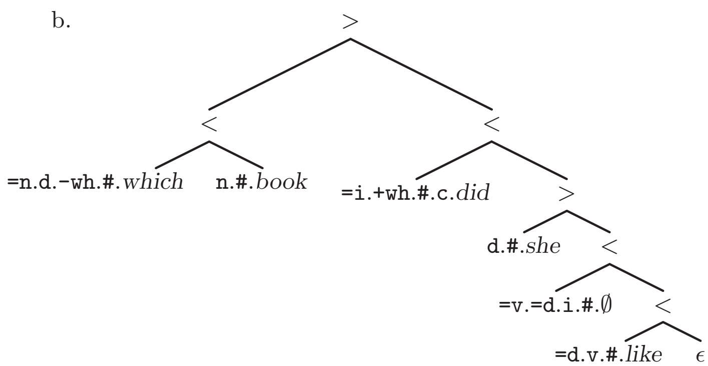
[7] Cf. (1) from above, and also Figure 3 from the appendix with ϕ instead of τ.

[8] Our treatment of relative clauses has been radically simplified for the sake of brevity. We take $\emptyset$ to denote a string of non–syntactic features without phonetic content.

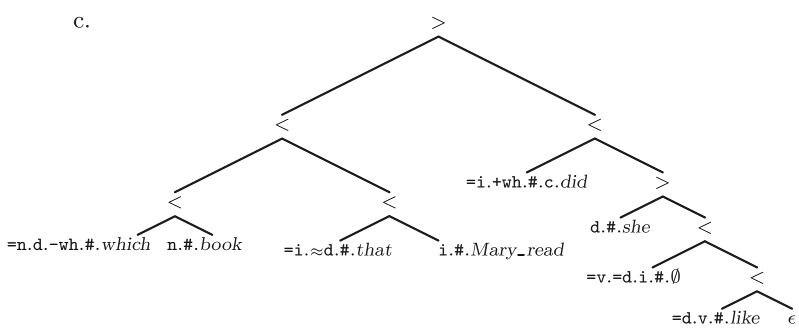
(10) a.



b.



c.



Note that the transition from (10b) to (10c) crucially involves availability of a checked feature instance, i.e. the instance of `d` on *which*, for *adjoin* to be able to apply to *which book did she like* "late." Resorting to this representational option is what distinguishes MG enriched by (countercyclic) *adjoin* from classical MGs, where checked features are radically inert and therefore deleted instantaneously.

7.2.3 Derivational Generative Capacity

Let us now turn to the question as to what the addition of late adjunction implies for the generative capacity of MGs. In the light of the type of example discussed earlier, there is no difference between early and late adjunction for finally resulting trees, or to put it differently, abstracting away from the binding phenomenon the system displays a “Church-Rosser-like” behavior. Therefore neither the weak nor the strong generative capacity of the formalism is affected by adding this type of late adjunction. Yet, this holds only insofar as the adjuncts do not introduce unchecked instances of (move–)licensees that allow subsequent extraction (out) of these adjuncts at some later derivation step. At the same time it is important to note that the kind of restriction required to enforce this actually yields welcome results, since without it we would be able to derive locality violations, such as shown in (11).

(11) *When$_i$ did John wonder [who$_j$ Mary met t_j t_i]

These considerations aside, what will under any circumstances be affected by the introduction of late adjunction is, what could — in the spirit of Becker et al. (1992) — be called the *derivational generative capacity*.[9] As already mentioned, our definition of *adjoin* crucially requires features not to be deleted even after they have been checked by an application of a structure building operation. In order to allow late(r) adjunction in full generality these features have to be present throughout the derivation.[10] In fact, this prevents us from adopting the methods which, in particular, led to the succinct, “chain–based” MG–reformulation (reducing MGs to their “bare essentials”) presented in Stabler and Keenan (2000). There, a minimalist tree is represented as a finite sequence of triples of finite strings such that only maximal subtrees with unchecked syntactic features are represented as components. More concretely, only those subtrees are represented which can still be operated on by the operations *merge* or *move*, and each of these subtrees is represented in a highly reduced form: indicating a) the unchecked syntactic features of the subtree’s head–label, b) whether the subtree has already a feature checked off or not (denoted by : and

[9]Note that, introducing their notion of derivational generative capacity, Becker et al. restrict their interest to predicate–argument relations, which they formally account for by means of a “coindexing” of predicates and their arguments. This coindexing is straightforwardly realized by assuming that a predicate and its arguments are introduced as dependent on each other, and such a dependence is initialized in a single derivation step. Of course, in this sense, an adjunct is not derivationally dependent on the constituent it modifies, and vice versa.

[10]This is true at least for plain categorial features.

::, respectively),[11] and c) the (narrow) non–syntactic yield of the subtree. Thus, MGs in exactly the sense of this succinct MG–reformulation can be seen as a severe reduction of "classical" MGs. This holds not only w.r.t. the strong generative capacity, but also the elusive notion of derivational complexity, since we are left with just one option for adjunction, namely, the earliest one. In other words, what is a familiar notational shortcut in syntactic representations (cf. item (9h) in our lexicon) becomes an essential part of the theory in the succinct MG–formulation. See (12a)–(12c), which correspond to (10a)–(10c) respectively.

(12) a. $\Big\langle \langle\, \texttt{+wh.c}\,, :, \textit{did\_she\_like}\,\rangle\,, \langle\, \texttt{-wh}\,, :, \textit{which\_book}\,\rangle \Big\rangle$

b. $\Big\langle \langle\, \texttt{c}\,, :, \textit{which\_book\_did\_she\_like}\,\rangle \Big\rangle$

c. $\Big\langle \langle\, \texttt{c}\,, :, \textit{which\_book\_that\_Mary\_read\_did\_she\_like}\,\rangle \Big\rangle$

Note that, as long as we do not permit late adjunction and restrict our interest to convergent derivations, there is a general finite upper bound on the number of components which must be available. But, to allow for unrestricted late(r) adjunction in such a representation, i.e. adjunction to any maximal projection at any stage of the derivation, an unbounded number of components must be essentially available.

One of the questions arising at this point is the following: when we take into account (only) the "Lebeaux cases" of late adjunction, is it then possible to finitely restrict the number of nodes to which late adjunction can apply without reducing too much the derivational capacity which seems to be necessary for an adequate description of the phenomena discussed? For example, is it possible to revise the definition of *adjoin* given above such that adjunction is only allowed to τ or one of its immediate daughters? The usefulness of such a restriction is straightforward, since it would (re)open the possibility for a succinct MG–formulation again, which in its turn makes the formalism directly amenable to polynomial–time parsing methods (see e.g. Harkema 2000).[12]

[11] That is, "::" serves to denote exactly the unaffected instances of lexical items.

[12] In a thematically related paper dealing with adjuncts in the TAG–framework, Schabes and Shieber (1994) suggest a modified notion of derivation, called *independent derivation*. Contrary to standard *dependent derivations* as defined in Vijay–Shanker (1987), this new mechanism effectively allows multiple adjunction at one and the same node ν. Thus, adjunction constraints valid at ν affect all constituents adjoined to ν irrespective of their ultimate hierarchical order in derived trees. The same effect is automatically captured in MGs as defined here and in Frey and Gärtner (2002), since (i) the MG–operation *adjoin* checks features against the head

7.3 Further Outlook

There are two obvious directions in which to pursue these issues further.[13]

First, it would be important to find out the consequences for the generative capacity of MGs with late adjunction that do not impose the restriction on adjuncts we looked at in Section 7.2.3. This is particularly interesting for MGs enriched with a mechanism for the treatment of relative clause extraposition. If, as we assume, the latter is analyzed in terms of a *(rightward) scrambling* operation, definable in analogy to its "leftward" counterpart from Frey and Gärtner (2002), we seem to be forced to lift the ban on multiple occurrences of competing *scramble–licensees* that was assumed there, following the corresponding constraint on move–licensees. Otherwise, there exist cases of multiple extraposition the structurally adequate derivation of which is possible in the relevant MGs *with* late adjunction but not in those MGs *without*. This means that these two types of MGs would differ not only in terms of their derivational generative capacity but also in terms of (at least) their strong generative capacity. (13) exemplifies such a case.

(13) [[[[Only those papers t_i]$_k$ did [everyone t_j] read t_k] [who was on the committee]$_j$] [which deal with adjunction]$_i$]

A step by step derivation of this example is provided in (14), where α and β are placeholders for *who was on the committee* and *which deal with adjunction*, respectively.

(14) ⋮

a. did [everyone] read [only those papers]
b. [only those papers]$_i$ did [everyone] read t_i
c. [only those papers]$_i$ did [[everyone] [α]] t_i read
d. [only those papers]$_i$ did [everyone t_j] t_i read [α]$_j$
e. [[only those papers] [β]]$_i$ did [everyone t_j] t_i read [α]$_j$
f. [only those papers t_k]$_i$ did [everyone t_j] t_i read [α]$_j$ [β]$_k$

Secondly, the issue of countercyclicity can be systematically further complicated by considering (a) countercyclic *move* and (b) counter-

of the constituent adjoined to, and (ii) the head of the constituent adjoined to is identical to the head of the resulting tree.

[13] As for semantics, it seems that the introduction of countercyclicity is neutral w.r.t. the question as to whether derivation trees or derived trees are interpreted. In case of the former, one would, in order to preserve compositionality, have to employ the "open–property–variable–approach" to restrictive relative clauses introduced by Bach and Cooper (1978) and discussed by Janssen (1982). (Thanks to Shalom Lappin for having raised this question.)

cyclic *merge* in addition. Countercyclic *move* seems to be necessary for the approach to Malagasy adverb placement by Rackowski and Travis (2000), in order to circumvent so-called "freezing violations," i.e. extraction from constituents that have undergone movement at an earlier stage (Thiersch p.c.). Collins (1994), however, provides arguments against this kind of approach. Likewise, it is easy to see that allowing countercyclic *merge* in addition to countercyclic adjunction would jeopardize the Lebeaux–account presented above. Nevertheless, from a formal point of view it would be attractive to find out whether there is a hierarchy in terms of derivational generative capacity ordering these different countercyclic operations.

Appendix

Throughout we let $\neg Syn$ and Syn be a finite set of *non–syntactic features* and a finite set of *syntactic features*, respectively, in accordance with (F1)–(F3) below. We take $Feat$ to be the set $\neg Syn \cup Syn$.

(F1) $\neg Syn$ is disjoint from Syn and partitioned into the sets $Phon$ and Sem, a set of *phonetic features* and a set of *semantic features*, respectively.

(F2) Syn is partitioned into five sets:[14]

$Base$		a set of *(basic) categories*
$M\text{-}Select$	$= \{ \mathtt{=x} \mid \mathtt{x} \in Base \}$	a set of *m(erge)–selectors*
$A\text{-}Select$	$= \{ \approx\mathtt{x} \mid \mathtt{x} \in Base \}$	a set of *a(djoin)–selectors*
$Licensees$	$= \{ \mathtt{-x} \mid \mathtt{x} \in Base \}$	a set of *licensees*
$Licensors$	$= \{ \mathtt{+x} \mid \mathtt{x} \in Base \}$	a set of *licensors*

(F3) $Base$ includes at least the category $\mathtt{c}$.

Definition 15 An *expression (over Feat)*, also referred to as a *minimalist tree (over Feat)*, is a 5–tuple $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle$ obeying (E1)–(E3).

(E1) $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ is a finite, binary (ordered) tree defined in the usual sense: N_τ is the finite, non–empty set of *nodes*, and $\triangleleft^*_\tau$ and $\prec_\tau$ are the respective binary relations of *dominance* and *precedence* on N_τ.[15]

(E2) $<_\tau \subseteq N_\tau \times N_\tau$ is the asymmetric relation of *(immediate) projection* that holds for any two siblings, i.e., for each $x \in N_\tau$

[14] Elements from Syn will usually be typeset in typewriter mode.

[15] Thus, $\triangleleft^*_\tau$ is the reflexive–transitive closure of $\triangleleft_\tau \subseteq N_\tau \times N_\tau$, the relation of *immediate dominance* on N_τ.

different from the root of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau\rangle$ either $x <_\tau$ $sibling_\tau(x)$ or $sibling_\tau(x) <_\tau x$ holds.[16]

(E3) $label_\tau$ is the *leaf–labeling function* from the set of leaves of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau\rangle$ into $Syn^*\{\texttt{\#}\}Syn^*Phon^*Sem^*$.[17]

We take $Exp(Feat)$ to denote the class of all expressions over $Feat$.

Let $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau\rangle \in Exp(Feat)$.[18]

For each $x \in N_\tau$, the *head of x (in τ)*, denoted by $head_\tau(x)$, is the (unique) leaf of τ with $x \triangleleft^*_\tau head_\tau(x)$ such that each $y \in N_\tau$ on the path from x to $head_\tau(x)$ with $y \neq x$ projects over its sibling, i.e. $y <_\tau sibling_\tau(y)$. The *head of* τ is the head of τ's root. τ is said to be a *head* (or *simple*) if N_τ consists of exactly one node, otherwise τ is said to be a *non–head* (or *complex*).

An $\upsilon = \langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon, <_\upsilon, label_\upsilon\rangle \in Exp(Feat)$ is a *subexpression of* τ in case $\langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon\rangle$ is a subtree of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau\rangle$, $<_\upsilon = <_\tau\restriction_{N_\upsilon \times N_\upsilon}$, and $label_\upsilon = label_\tau\restriction_{N_\upsilon}$. Such a subexpression υ is a *maximal projection (in* τ*)* if its root is a node $x \in N_\tau$ such that x is the root of τ, or such that $sibling_\tau(x) <_\tau x$. $MaxProj(\tau)$ is the set of maximal projections in τ.

An $\upsilon \in MaxProj(\tau)$ is said to *have*, or *display*, *(open) feature* f if the label assigned to υ's head by $label_\tau$ is of the form $\beta\texttt{\#}f\beta'$ for some $f \in Feat$ and some $\beta, \beta' \in Feat^*$.[19]

τ is *complete* if its head–label is in $Syn^*\{\texttt{\#}\}\{\texttt{c}\}Phon^*Sem^*$, and each of its other leaf–labels is in $Syn^*\{\texttt{\#}\}Phon^*Sem^*$. Hence, a complete expression over *Feat* is an expression that has category `c`, and this instance of `c` is the only instance of a syntactic feature within all leaf–labels which is preceded by an instance of `#`.

The *phonetic yield of* τ, denoted by $Y_{Phon}(\tau)$, is the string which results from concatenating in "left–to–right–manner" the labels assigned via $label_\tau$ to the leaves of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau\rangle$, and replacing all instances of non–phonetic features with the empty string, afterwards.

---

[16] $sibling_\tau(x)$ denotes the (unique) sibling of any given $x \in N_\tau$ different from the root of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau\rangle$. If $x <_\tau y$ for some $x, y \in N_\tau$ then x is said to *(immediately) project over y*.

[17] For each set M, M^* is the Kleene closure of M, including ϵ, the empty string. For any two sets of strings, M and N, MN is the product of M and N w.r.t. string concatenation. Further, `#` denotes a new symbol not appearing in *Feat*.

[18] Note that the leaf–labeling function $label_\tau$ can easily be extended to a total labeling function ℓ_τ from N_τ into $Feat^*\{\texttt{\#}\}Feat^* \cup \{\texttt{<}, \texttt{>}\}$, where `<` and `>` are two new distinct symbols: to each non–leaf $x \in N_\tau$ we can assign a label from $\{\texttt{<}, \texttt{>}\}$ by ℓ_τ such that $\ell_\tau(x) = \texttt{<}$ iff $y <_\tau z$ for $y, z \in N_\tau$ with $x \triangleleft_\tau y, z$, and $y \prec_\tau z$. In this sense a concrete $\tau \in Exp(Feat)$ is depictable in the way done in (10a)–(10c).

[19] Thus, e.g., the expression depicted in (10a) has feature `+wh`, while there is a maximal projection which has feature `-wh`.

For any $\upsilon, \phi \in \mathit{Exp}(\mathit{Feat})$, $[_<\upsilon, \phi\,]$ (respectively, $[_>\upsilon, \phi\,]$) denotes the complex expression $\chi = \langle N_\chi, \triangleleft^*_\chi, \prec_\chi, <_\chi, \mathit{label}_\chi \rangle \in \mathit{Exp}(\mathit{Feat})$ for which υ and ϕ are those two subexpressions such that $r_\chi \triangleleft_\chi r_\upsilon$, $r_\chi \triangleleft_\chi r_\phi$ and $r_\upsilon \prec_\chi r_\phi$, and such that $r_\upsilon <_\chi r_\phi$ (respectively $r_\phi <_\chi r_\upsilon$), where r_υ, r_ϕ and r_χ are the roots of υ, ϕ and χ, respectively.

For any $\upsilon, \phi, \chi \in \mathit{Exp}(\mathit{Feat})$ such that ϕ is a subexpression of υ, $\upsilon\{\phi/\chi\}$ is the expression which results from substituting χ for ϕ in υ.

Definition 16 A *minimalist grammar with generalized adjunction (*abbr. *MG^{adj})* is a five–tuple $G = \langle \neg \mathit{Syn}, \mathit{Syn}, \mathit{Lex}, \Omega, \mathtt{c} \rangle$ with Ω being the operator set consisting of the structure building functions *merge*, *move* and *adjoin* defined w.r.t. *Feat* as in (me), (mo) and (ad) below, respectively, and with *Lex* being a *lexicon (over Feat)*, i.e., *Lex* is a finite set of simple expressions over *Feat*, and each lexical item $\tau \in \mathit{Lex}$ is of the form $\langle \{r_\tau\}, \triangleleft^*_\tau, \prec_\tau, <_\tau, \mathit{label}_\tau \rangle$ such that $\mathit{label}_\tau(r_\tau)$ is an element from $\{\mathtt{\#}\}(\mathit{M\text{-}Select} \cup \mathit{Licensors})^*(\mathit{Base} \cup \mathit{A\text{-}Select})\,\mathit{Licensees}^*\,\mathit{Phon}^*\,\mathit{Sem}^*$.

The operators from Ω build larger structure from given expressions by succesively checking "from left to right" the instances of syntactic features appearing within the leaf–labels of the expressions involved. The symbol # serves to mark which feature instances have already been checked by the application of some structure building operation.

(me) *merge* is a partial mapping from $\mathit{Exp}(\mathit{Feat}) \times \mathit{Exp}(\mathit{Feat})$ into $\mathit{Exp}(\mathit{Feat})$. For any $\upsilon, \phi \in \mathit{Exp}(\mathit{Feat})$, $\langle \upsilon, \phi \rangle$ is in $\mathit{Domain}(\mathit{merge})$ if for some category $\mathtt{x} \in \mathit{Base}$ and $\alpha, \alpha', \beta, \beta' \in \mathit{Feat}^*$, conditions (i) and (ii) are fulfilled:

(i) the head–label of υ is $\alpha\mathtt{\#{=}x}\alpha'$ (i.e. υ has m–selector $\mathtt{=x}$), and
(ii) the head–label of ϕ is $\beta\mathtt{\#x}\beta'$ (i.e. ϕ has category $\mathtt{x}$).

Then,

(me.1) $\mathit{merge}(\upsilon, \phi) = [_<\upsilon', \phi'\,]$ if υ is simple, and
(me.2) $\mathit{merge}(\upsilon, \phi) = [_>\phi', \upsilon'\,]$ if υ is complex,

where υ' and ϕ' result from υ and ϕ, respectively, just by interchanging the instance of # and the instance of the feature directly following the instance of # within the respective head–label (cf. Fig. 1).

(mo) *move* is a partial mapping from $\mathit{Exp}(\mathit{Feat})$ into $\mathit{Exp}(\mathit{Feat})$. An $\upsilon \in \mathit{Exp}(\mathit{Feat})$ is in $\mathit{Domain}(\mathit{move})$ if for some $\mathtt{-x} \in \mathit{Licensees}$ and $\alpha, \alpha' \in \mathit{Feat}^*$, (i) and (ii) are true:

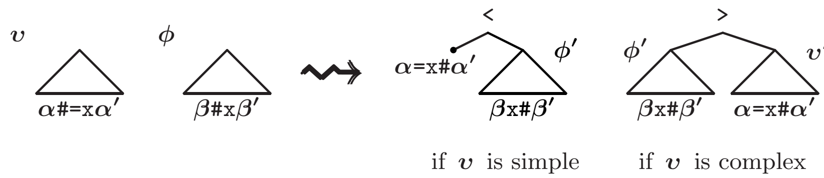


FIGURE 1 $merge(\upsilon, \phi)$ according to (me).

(i) the head–label of υ is α`#+x`α' (i.e. υ has licensor `+x`), and
(ii) there is exactly one $\phi \in \mathit{MaxProj}(\upsilon)$ with head–label β`#-x`β' for some $\beta, \beta' \in \mathit{Feat}^*$ (i.e. there is exactly one $\phi \in \mathit{MaxProj}(\upsilon)$ displaying `-x`).

Then,

$$\mathit{move}(\upsilon) = [_{>}\phi', \upsilon'],$$

where $\upsilon' \in \mathit{Exp}(\mathit{Feat})$ results from υ by interchanging the instance of `#` and the instance of `+x` directly following it within head–label of υ, while the subtree ϕ is replaced by a single node labeled ϵ. $\phi' \in \mathit{Exp}(\mathit{Feat})$ arises from ϕ by interchanging the instance of `#` and the instance of `-x` next to its right within the head–label of ϕ (cf. Fig. 2).

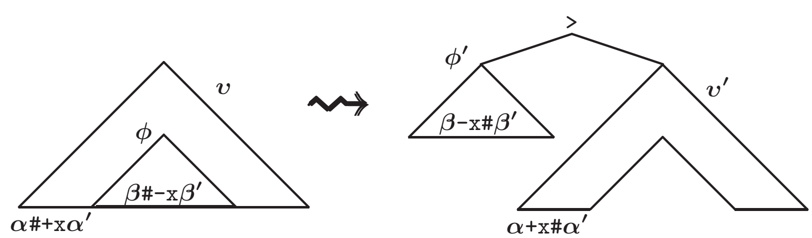


FIGURE 2 $\mathit{move}(\upsilon)$ according to (mo).

(ad) *adjoin* is a partial mapping from $\mathit{Exp}(\mathit{Feat}) \times \mathit{Exp}(\mathit{Feat})$ to $\mathcal{P}_{\text{fin}}(\mathit{Exp}(\mathit{Feat}))$.[20] A pair $\langle \upsilon, \phi \rangle$ with $\upsilon, \phi \in \mathit{Exp}(\mathit{Feat})$ belongs to $\mathit{Domain}(\mathit{adjoin})$ if for some category $\mathtt{x} \in \mathit{Base}$ and $\alpha, \alpha' \in \mathit{Feat}^*$, conditions (i) and (ii) are fulfilled:

(i) the head–label of υ is α`#≈x`α' (i.e. υ has a–selector `≈x`), and
(ii) there is some $\chi \in \mathit{MaxProj}(\phi)$ with head–label β`#x`β' or β`x`β'`#`β'' for some $\beta, \beta', \beta'' \in \mathit{Feat}^*$

[20] $\mathcal{P}_{\text{fin}}(\mathit{Exp}(\mathit{Feat}))$ is the class of all finite subsets of $\mathit{Exp}(\mathit{Feat})$.

Then,

$$adjoin(\upsilon,\phi) = \left\{ \phi\{\chi/[_<\chi,\upsilon']\} \;\middle|\; \begin{array}{l}\chi \in \mathit{MaxProj}(\phi) \text{ with head–} \\ \text{label } \beta\texttt{\#x}\beta' \text{ or } \beta\texttt{x}\beta'\texttt{\#}\beta'' \text{ for} \\ \text{some } \beta,\ \beta',\ \beta'' \in \mathit{Feat}^*\end{array} \right\},$$

where υ' results from υ by interchanging the instances of # and $\approx$x, the latter directly following the former within the head–label of υ. (cf. Fig. 3).

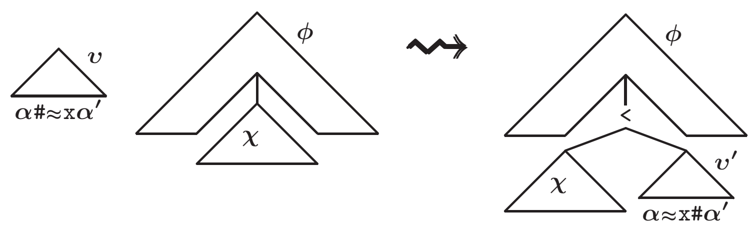


FIGURE 3 Expression from $adjoin(\upsilon,\phi)$ according to (ad).

For each MG$^{\text{adj}}$ $G = \langle \neg Syn, Syn, Lex, \Omega, \mathsf{c}\rangle$, the *closure of* G, $CL(G)$, is the set $\bigcup_{k\in\mathbb{N}} CL^k(G)$,[21] where $CL^0(G) = Lex$, and for $k \in \mathbb{N}$, $CL^{k+1}(G) \subseteq Exp(Feat)$ is recursively defined as the set

$$\begin{aligned} & CL^k(G) \\ & \cup \{merge(\upsilon,\phi) \,|\, \langle\upsilon,\phi\rangle \in Domain(merge) \cap CL^k(G) \times CL^k(G)\} \\ & \cup \{move(\upsilon) \,|\, \upsilon \in Domain(move) \cap CL^k(G)\} \\ & \cup \bigcup\nolimits_{\langle\upsilon,\phi\rangle \in Domain(adjoin) \cap CL^k(G)\times CL^k(G)} adjoin(\upsilon,\phi) \end{aligned}$$

The set $\{\tau \,|\, \tau \in CL(G) \text{ and } \tau \text{ complete}\}$, denoted by $T(G)$, is the *minimalist tree language derivable by* G. The set $\{Y_{Phon}(\tau) \,|\, \tau \in T(G)\}$, denoted by $L(G)$, is the *minimalist (string) language derivable by* G.

References

Bach, E. and R. Cooper. 1978. The NP–S analysis of relative clauses and compositional semantics. *Linguistics and Philosophy* 2:145–150.

Becker, T., O. Rambow, and M. Niv. 1992. The derivational generative power of formal systems or Scrambling is beyond LCFRS. Technical Report IRCS-92-38, Institute for Research in Cognitive Science, University of Pennsylvania.

Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

[21] $\mathbb{N}$ is the set of all non–negative integers.

Chomsky, N. 1995. *The Minimalist Program*. Cambridge: MIT Press.

Chomsky, N. 2000. Minimalist inquiries. The framework. In R. Martin, D. Michaels, and J. Uriagereka, eds., *Step by Step. Essays on Minimalist Syntax in Honor of Howard Lasnik*, pages 89–155. Cambridge: MIT Press.

Collins, C. 1994. Economy of derivation and the Generalized Proper Binding Condition. *Linguistic Inquiry* 25:45–61.

Frey, W. and H.-M. Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of the Conference on Formal Grammar*, pages 41–52.

Harkema, H. 2000. A recognizer for minimalist grammars. In *Proceedings of the 6th International Workshop on Parsing Technologies*, pages 111–122.

Janssen, T. M. V. 1982. Compositional semantics and relative clause formation in Montague grammar. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, eds., *Formal Methods in the Study of Language*, pages 237–276. Dordrecht: Foris.

Joshi, A. K. and Y. Schabes. 1997. Tree adjoining grammars. In G. Rozenberg and A. Salomaa, eds., *Handbook of Formal Languages*, vol. 3, pages 69–124. Berlin, Heidelberg: Springer.

Lebeaux, D. 1991. Relative clauses, licensing, and the nature of the derivation. In S. D. Rothstein, ed., *Perspectives on Phrase Structure. Heads and Licensing*, pages 209–239. New York: Academic Press.

Michaelis, J. 2001. Derivational minimalism is mildly context–sensitive. In M. Moortgat, ed., *Logical Aspects of Computational Linguistics, LNAI 2014*, pages 179–198. Berlin: Springer. Also available at `http://www.ling.uni- potsdam.de/~michael/papers.html`.

Rackowski, A. and L. Travis. 2000. V–initial languages. X or XP movement and adverbial placement. In A. Carnie and E. Guilfoyle, eds., *The Syntax of Verb Initial Languages*, pages 117–141. New York: Oxford University Press.

Schabes, Y. and S. M. Shieber. 1994. An alternative conception of tree–adjoining derivation. *Computational Linguistics* 20:91–124.

Stabler, E. P. 1997. Derivational minimalism. In C. Retoré, ed., *Logical Aspects of Computational Linguistics, LNAI 1328*, pages 68–95. Berlin: Springer.

Stabler, E. P. 1999. Remnant movement and complexity. In G. Bouma, G. M. Kruijff, E. Hinrichs, and R. T. Oehrle, eds., *Constraints and Resources in Natural Language Syntax and Semantics*, pages 299–326. Palo Alto: CSLI Publications.

Stabler, E. P. 2001. Recognizing head movement. In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics, LNAI 2099*, pages 245–260. Berlin: Springer.

Stabler, E. P. and E. L. Keenan. 2000. Structural similarity. In *Proceedings of the 16th Twente Workshop on Language Technology (TWLT 16) joint with the 2nd AMAST Workshop on Algebraic Methods in Language Processing*, pages 251–265. Iowa City.

Vijay–Shanker, K. 1987. *A Study of Tree Adjoining Grammars*. Ph.D. thesis, University of Pennsylvania.